

# Large Language Model을 통한 대화 데이터셋 자동 생성 및 검색 성능 향상

최형준<sup>o,1</sup>, 홍범석<sup>2</sup>, 최원석<sup>2</sup>, 한영섭<sup>2</sup>, 전병기<sup>2</sup>, 나승훈<sup>1</sup>  
<sup>1</sup>전북대학교, <sup>2</sup>LG 유플러스

modera2017@gmail.com, {bshong, wseokchoi, yshan042, bkjeon}@lguplus.co.kr, nash@jbnu.ac.kr

## Conversation Dataset Generation and Improve Search Performance via Large Language Model

Hyeongjun Choi<sup>o,1</sup>, Beomseok Hong<sup>2</sup>, Wonseok Choi<sup>2</sup>, Youngsub Han<sup>2</sup>, Byoung-Ki Jeon<sup>2</sup>, Seung-Hoon Na<sup>1</sup>  
<sup>1</sup>Jeonbuk National University, <sup>2</sup>LG Uplus

### 요약

대화 데이터와 같은 데이터는 사람이 수작업으로 작성해야 하기 때문에 데이터셋 구축에 시간과 비용이 크게 발생한다. 현재 대두되고 있는 Large Language Model은 이러한 대화 생성에서 보다 자연스러운 대화 생성이 가능하다는 이점이 존재한다. 이번 연구에서는 LLM을 통해 사람이 만든 적은 양의 데이터셋을 Fine-tuning 하여 위키백과 문서로부터 데이터셋을 만들어내고, 이를 통해 문서 검색 모델의 성능을 향상시켰다. 그 결과 학습 데이터와 같은 문서집합에서 MRR 3.7%p, 위키백과 전체에서 MRR 4.5%p의 성능 향상을 확인했다.

**주제어:** 데이터 증강, LLM, 문서 검색

### 1. 서론

언어모델의 학습을 위해서는 고품질의 데이터를, 대량으로 수집하는 것이 중요하다. 하지만 이러한 고품질의 데이터를 획득하는 것은 어려운 문제이고, 특히 문서에 기반하여 대화형태의 질의 응답을 구성하기 위한 데이터셋은 수작업으로 데이터셋을 만들어 내야 하기 때문에 비용과 시간이 매우 크게 소요된다. 최근에 대두되고 있는 Large Language Model은 Fine-tuning이 없더라도 자연스러운 자연어 생성 능력이 우수한 것으로 알려져 있다. LLM을 통해서 생성한 문장은 기존의 작은 모델에 비해 우수한 품질을 가지기 때문에 이를 통해 데이터 증강을 수행 할 경우, 인간이 직접 작성한 데이터셋에 비해서 낮은 품질이겠지만 충분히 높은 품질의 데이터를 적은 비용으로 대량 생성하는 것이 가능하다.

현재의 LLM은 생성 목표에 대해 아무런 학습 및 지식 없이 적용되는 Zero-shot 환경에서 높은 가용성을 보여주고 있지만, 인간과 같이 몇 개의 데이터를 통해 전체 데이터를 유추하는 것을 완벽하게 수행 할 수는 없다. 하지만 데이터 증강과 같이 특정한 영역에서 국한하여 생성해야 하는 경우에는 Fine-tuning을 적용하여 모델이 해당 데이터에 적응하도록 할 수 있다. Alpaca[1]나 GPT4All[2]의 사례를 고려하면 Zero-shot 환경에 비해 Fine-tuning이 가지는 이점은 매우 강력하다고 볼 수 있다.

이번 연구에서는 QLoRA[3]를 통해 경량화된 환경에서 Polyglot[4]모델에 인간이 작성한 대화 데이터셋을 통해 Fine-tuning을 적용한 후, 이를 통해 위키백과 기반 대화 데이터셋을 생성하여 Passage 검색 모델의 성능을 향상 시킬 수 있음을 확인했다. Fine-tuning 된 모델을 통해 위키백과 문서의 약 50%

에 해당하는 11여만개의 문서를 통해 160여만개의 대화 턴을 생성했고, 이를 통해 사전학습을 한 후 최종적으로 인간이 작성한 데이터셋을 통해 Fine-tuning을 한 결과 학습 데이터와 같은 문서집합에서 MRR 3.7%p, 위키백과 전체에서 MRR 4.5%p의 성능 향상을 확인했다.

### 2. 관련 연구

#### 2.1 LLM을 통한 데이터 증강

LLM을 통해 생성된 데이터로 보다 작은 모델의 성능을 향상 시킬 수 있다는 것은 이전에 보여진 바가 있다. Alpaca[1]는 LLaMA[5]모델에 대해 ChatGPT를 통해 생성한 Instruction 데이터셋을 통해 LLaMA 모델을 Fine-tuning하여 기존의 모델에서 부족한 Instruction-following 성능을 향상 시켰을 뿐 아니라 ChatGPT와 유사한 성능을 더 작은 모델에서 낼 수 있음을 확인 했다. GPT4All[2]역시 ChatGPT를 통해 생성된 데이터를 여러 단계의 필터링을 통해 정제, LoRA 등의 최적화 기법을 적용하여 개인용 PC 수준에서도 ChatGPT와 유사한 성능을 낼 수 있는 언어모델을 구축하는 방법을 제시했다.

#### 2.2 경량화된 Fine-tuning

LLM은 일반적으로 10억개 이상의 파라미터를 가지고 있기 때문에 이를 완전히 Fine-tuning 할 경우 매우 큰 비용이 발생한다. 그렇기 때문에 이 과정을 경량화 하는 방법은 다양하게 제시되어 왔다. Adapter 레이어[6]는 모델의 각 레이어마다 학습 가능한 파라미터를 가진 Adapter를 추가하여 레이어의 입력과 출력에 대해 수정을 적용한다. Low-rank Adaptation[7]는 기존의 Adapter 레이어가 모델의 입력과 출력에 직렬적으로 존재

하기 때문에 생기는 비효율을 해결하기 위해 제시된 방법으로, Low-rank decomposition을 통해 각 레이어와 병렬적으로 존재하는 가중치를 추가하여 연산의 효율을 향상 시키고 기존 모델의 1% 미만의 가중치로 Fine-tuning을 적용 할 수 있도록 했다. (IA)<sup>3</sup>[8]은 LoRA보다 더 적은 가중치를 통해 Fine-tuning 하는 방법으로, Transformer[9] 구조에서 어텐션의 Key와 Value, Feed-forward Network에 rescaling을 적용한다.

이러한 Adapter layer는 기반이 되는 모델의 계산 과정에서 간섭하지 않기 때문에 모델 양자화를 통해 기반 모델의 연산을 경량화 한 후, 이에 Adaptation을 적용 할 수도 있다. QLoRA[3]는 모델의 양자화와 LoRA를 동시에 적용하여 더 작은 규모의 장치에서 모델을 학습하면서 거의 동등한 성능을 낼 수 있음을 확인했다.

### 3. 데이터 구성

#### 3.1 데이터셋

실험을 위해 사용한 데이터셋은 LG U+에서 제공한 사람이 작성한 문서 기반 한국어 대화 데이터셋을 사용했다. 이는 한국어 위키에 기반하여 사용자와 에이전트 간의 대화로 구성되어 있고, 사람이 직접 작성했다. 각 대화 데이터는 8턴 가량의 사용자와 에이전트간 대화와, 해당 대화의 주제가 되는 하나의 위키백과 문서, 그리고 각 에이전트의 답변의 근거가 되는 문서 상의 Span이 주어진다. 이때 답변의 근거를 Reference라고 한다. 문서에 기반하지 않은 일반적인 응답의 경우 Reference는 포함되지 않는다. 자연스러운 연속 대화를 생성하기 위해 데이터셋 생성 모델을 학습하는 과정에서는 이를 사용했지만, 검색 모델을 학습 할 때는 불필요하기 때문에 이를 제외했다. 데이터셋에 포함된 대화는 총 5,281개이고, 이중 4,752개를 학습 데이터, 529개를 평가 데이터로 사용했다. 이 데이터셋에 포함된 위키백과 문서는 3,238개 이고, 총 15,324개의 Passage로 분할된다.

#### 3.2 위키백과 전처리

데이터셋 생성 및 검색 모델의 학습을 위해 한국어 위키백과의 모든 문서를 사용했다. 많은 수의 문서가 매우 긴 길이를 가지고 있고, 매우 긴 길이를 가지는 문서는 이를 통해 응답 생성을 하는 등의 활용을 하기에 비효율적이기 때문에 적절한 길이를 가지는 Passage 단위로 문서를 분할했다. 문서를 분할할 때 각 Passage 내의 내용이 어느정도 일관성을 가질 수 있도록 위키백과의 섹션을 기준으로 분할 했다. 이러한 섹션은 하위 섹션을 가질 수 있고, 이는 다시 하위 항목으로 분할 될 수 있다. 이번 실험에서는 차상위 섹션을 하나의 Passage로 분할되는 최소 단위로 사용했다. 이때 분리된 Passage의 길이가 긴 경우, 각 부분이 최대한 길이가 같도록 문장 단위로 분할하여



그림 1. 한국어 위키 문서의 분할 과정

새로운 Passage로 나눴다. 이때 분할하는 기준은 Tokenizer를 기준으로 512토큰으로, 이보다 긴 경우 다른 Passage가 되도록 했다. 또, 하위 섹션의 길이가 짧아 하나의 Passage로 병합하는 것이 효율적인 경우가 존재하기 때문에 하위 섹션의 토큰 길이의 합이 512토큰 미만인 경우 이를 하나로 병합했다. 그림 1은 이러한 과정의 예시이다. 사용된 위키백과 문서는 총 232,973개이고 이는 562,076개의 Passage로 분할되었다.

### 4. 모델 구성

이번 실험에서는 데이터 생성 모델과 검색 모델, 응답 생성 모델 세 가지로 구성하여 진행했다. 데이터 생성 모델은 인간이 작성한 데이터셋을 통해 학습하여 입력 문서에 기반된 대화를 생성하는 모델로, 검색 모델의 학습을 위한 데이터 증강에 사용된다. 검색 모델은 응답 생성 시 선행되는 모델로, 답변을 생성하기 위해 필요한 근거가 되는 Passage를 전체 Passage 집합으로부터 찾아내는 모델이다. 마지막으로 응답 생성 모델은 사용자의 질의에 대한 답변을 생성하는 모델이다.

#### 4.1 데이터 생성 모델

데이터 생성 모델은 검색 모델의 학습을 위한 데이터를 확장하기 위해 사용된다. 일반적으로 검색 모델을 학습하기 위해서는 전체 문서 집합을 대상으로 대량의 데이터셋을 통해 학습을 진행해야 한다. 하지만, 대화 기반 질의 응답과 같은 작업에서는 이러한 대량의 데이터셋을 확보하는 것은 큰 비용을 요구하게

된다. 따라서, 인간이 데이터셋을 작성한다면 높은 품질의 데이터셋을 구축할 수 있겠지만, 검색의 대상이 되는 문서 집합이 방대할 경우 구축된 데이터셋은 문서 집합을 학습하기에는 매우 작은 규모가 되거나 매우 큰 비용을 초래할 수 있다.

이번 연구에서는 적은 양의, 인간을 통해 작성된 높은 품질의 데이터셋을 통해 Large Language Model에 Fine-tuning을 적용, 문서 집합 내의 문서에 기반해 대화 데이터셋을 생성하는 것을 목표로 했다. 이러한 과정을 통해 생성된 데이터셋은 인간을 통해 작성된 수준의 높은 품질을 가지지 않더라도 상대적으로 적은 비용을 통해 대량의 데이터셋을 생성할 수 있다. 이를 위해 공개된 한국어 LLM 중 하나인 Polyglot 12.8B 모델[4]을 사용했다. 이 모델은 GPT NeoX[10] 구조를 사용하여 한국어 블로그 포스트나 뉴스와 같은 말뭉치를 통해 사전 학습된 언어 모델로, 기존의 공개된 한국어 언어모델에 비해 높은 성능을 보였다.

데이터 생성 모델의 생성 목표는 Passage와 이전 대화 기록으로부터 다음 사용자와 에이전트 간의 대화를 생성하는 것이다. 이때, 에이전트의 대화를 생성하기 전에 Reference를 예측한 후 에이전트의 응답을 이어서 생성하도록 하는 Chain-of-Thought[11]을 적용했다. 일반적으로 Chain-of-Thought로 생성되는 근거는 이후 이어지는 응답에 대한 생성 품질을 향상시키기 위해 사용되지만, 이번 실험에서는 이에 더하여 데이터셋 생성 시 현재 사용중인 Passage를 얼마나 사용했는지 확인하기 위해 사용했다.

#### 4.1.1 경량화된 Fine-tuning

언어모델의 효율적인 Fine-tuning을 위해 Low-Rank Adaption(LoRA)[7]를 적용했다. Large Language Model을 직접 Fine-tuning할 경우 매우 큰 연산자원과 기억공간을 요구한다. 이는 모델 내의 모든 가중치에 대해 연산 후 업데이트를 진행해야 하기 때문이다. LoRA는 이러한 문제점에 대한 해결책 중 하나로, 모델의 각 레이어마다 학습 가능한 Rank decomposition 어댑터를 추가한다. 기존의 사전학습된 모델은 가중치를 고정시켜 학습시키지 않고 Low-rank로 분할된 가중치만 학습되도록 한 후, 이 레이어의 출력을 모델의 출력에 합산하는 방법으로 전체 모델을 학습가능하게 수정한다. 이때, Low-rank decomposition이 적용된 학습 가능한 레이어는 기존 레이어에 비해 상당히 작은 크기를 가지고 있기 때문에 메모리 및 연산 자원의 소요가 큰폭으로 줄어들면서 전체 모델을 Fine-tuning 한 것과 비슷한 효과를 낼 수 있다.

이러한 Adapter 레이어는 기존의 모델에 대해서는 각 레이어의 출력만 계산하면 되기 때문에 모델을 양자화 한 후, 이를 통해 연산된 출력에 LoRA 어댑터의 출력을 합산하는 방법으로 최적화를 추가적으로 적용할 수 있다[3]. 모델의 양자화는

int8와 같은 작은 크기의 자료형을 사용하여 연산을 수행한다. 모델 양자화는 가중치 벡터 내의 원소들을 int8과 같은 작은 크기의 자료형의 최대값과 최소값에 매핑시키는 방법이다.

이번 연구에서는 모델의 양자화와 LoRA를 적용, Fine-tuning 과정을 경량화 하여 학습을 진행하였다. 모델 양자화는 int8를 사용하여 양자화를 적용했다.

## 4.2 검색 모델

생성 모델을 통한 데이터 증강의 효과를 확인하기 위해 구성된 검색모델은 Dense Passage Retriever[12]을 사용했다. DPR은 BERT[13]와 같은 사전학습 언어모델을 통해 질의와 문서를 인코딩한 후, 두 표상 사이의 유사도를 계산하여 전체 Passage 집합으로부터 필요한 Passage를 검색하는 모델이다. 이 모델은 질의와 문서는 각각 개별적으로 모델을 통해 인코딩하는 Dual-encoder 구조의 모델이고 유사도를 계산해 벡터 내적을 통해 수행하기 때문에 전체 Passage 집합을 미리 인코딩한 후 빠르게 질의로부터 검색 결과를 도출할 수 있어 대규모 문서 집합으로부터 필요한 지식을 찾아내야 하는 Open-domain QA와 같은 Task에 적합한 검색 모델이다.

이번 연구에서는 기존 DPR의 학습 방법은 In-batch negative를 사용하는 것을 수정해서 같은 양의 Random-negative를 섞어 사용했다. 이는 위키백과 내에 포함되어 있는 Passage의 규모에 비해 인간이 작성한 데이터셋의 분량이 적기 때문에 In-batch negative만으로는 Negative로 사용할 수 있는 Passage가 매우 적어지는 문제가 존재하기 때문이다. 이때, Random-negative는 각 질의마다 하나씩 추가하여 사용했다. 데이터 생성 모델을 통해 학습하는 과정은 생성된 데이터를 통해 먼저 Pretraining을 적용한 후, 인간이 작성한 데이터를 통해 최종적으로 Fine-tuning하는 식으로 적용했다.

## 5. 실험

### 5.1 데이터 생성

검색 모델을 학습하기 위한 데이터 생성은 임의로 선택한 위키백과 문서 내의 Passage를 순차적으로 입력하여 생성하도록 했다. 이때, 문서 내의 각 Passage는 생성된 에이전트의 대화 중 근거 부분을 비교하여 Passage의 각 문장을 비교, 답변 생성에 사용된 문장이 전체 문장의 50%가 넘을 경우 다음 순서의 Passage를 사용하도록 했다.

데이터 생성 모델의 품질을 향상시키기 위해 생성된 대화 데이터에 대해 필터링을 적용했다. 필터링은 생성된 대화가 이전의 대화를 방지하거나, 기반이 되는 Passage의 내용을 단순히 복사하는 경우를 방지하기 위해 적용된다. 필터링을 적용하기 위한 점수는 생성된 대화를 Tokenizer를 통해 분할 시킨 후 Token-level Recall을 계산했다. 이는 생성된 대화의 토큰 중

얼마나 많은 토큰이 비교 대상 안에 포함되어 있는 지에 대한 비율로 아래와 같이 계산한다

$$R(p, g) = \frac{|\{t_p | t_p \in p, t_p \in g\}|}{|\{t_g | t_g \in g\}|}$$

필터링 되는 조건은 아래와 같다

- 생성된 사용자 질의나 에이전트의 응답이 이전 대화 기록과의 Recall이 0.5 이상인 경우
- 생성된 사용자 질의와 에이전트의 응답 사이의 Recall이 0.7 이상인 경우
- 생성된 에이전트의 응답과 예측한 Reference 사이의 Recall이 0.7 이상, 0.3 미만인 경우

이러한 경우 생성이 부적절하게 된 것으로 보고 현재의 Passage를 건너 뛰고 다음 Passage를 통해 생성을 이어나가도록 했다. 실험을 위해 생성한 대화 데이터는 총 117,329개의 문서를 통해 구성되었고, 이는 전체 문서의 50%정도에 해당한다. 생성된 톰 수는 평균 14.3으로 전체 1,677,843개의 톰을 생성했다. 표 3은 해당 대화주제를 통해 생성한 대화의 예시이다.

## 5.2 모델설정

문서기반 대화 데이터셋을 생성하기 위해 Fine-tuning을 적용한 기반 모델은 Polyglot 12.8B<sup>1</sup> 모델을 사용했다. Fine-tuning을 위한 LoRA Adapter는 Alpha 32, Rank 32, Dropout은 5%로 설정하여 학습을 진행했다. 검색 모델은 한국어 RoBERTa-base<sup>2</sup> 모델에 기반하여 DPR을 구성하여 학습했다.

## 6. 실험 결과

Method	In-domain	Pretrained	Fine-tuned
MRR	53.36	41.55	57.06
R@1	41.62	31.54	43.81
R@5	68.35	53.73	74.62
R@10	75.57	62.07	81.85

표 1. In-domain 검색모델 실험 결과

표 1은 In-domain 문서집합에서의 검색모델의 성능이다. In-domain 문서 집합은 위키백과 전체가 아니라 LG 데이터셋을 구축할 때 사용된 문서들에 대해서만 검색을 수행한 것이다. 생성된 데이터만 사용해서 사전학습을 먼저 적용했을 때 In-domain 문서집합만을 사용하여 학습했을 때 보다는 낮지만 어느정도의 성능을 낼 수 있음을 확인했고, 이후 전체 데이터를 통해 fine-tuning을 적용하면 기존에 비해 MRR 지표에서 3.7%p 향상되었다.

<sup>1</sup><https://huggingface.co/EleutherAI/polyglot-ko-12.8b>

<sup>2</sup><https://huggingface.co/klue/roberta-base>

Method	In-domain	Pretrained	Fine-tuned
MRR	40.48	43.39	45.03
R@1	28.73	30.89	32.77
R@5	53.62	57.71	60.84
R@10	65.33	67.28	68.68

표 2. Out-domain 검색모델 실험 결과

표 2는 Out-domain 문서집합에서의 검색모델의 성능이다. Out-domain 문서 집합은 위키백과 전체를 포함하는 문서집합으로, 거의 대부분 LG 데이터셋에 포함되지 않은 문서들로 구성되어 있다. In-domain 문서집합만을 사용하여 학습했을 때 생성한 데이터만 사용하여 사전학습 한 모델보다 낮은 검색 성능이 확인되었고, Fine-tuning까지 적용할 경우 기존 모델보다 MRR 지표에서 4.5%p 향상되었다.

## 7. 결론

이번 연구에서는 LLM을 인간이 작성한 대화 데이터셋을 통해 Fine-tuning하여 위키백과 문서로부터 대화 데이터셋을 생성하도록 하여 검색 모델의 성능을 향상시킬 수 있는지를 확인하였다. 그 결과 생성된 데이터셋을 사용하지 않았을 때와 비교하여 In-domain에서 학습 데이터와 같은 문서집합에서 MRR 3.7%p, 위키백과 전체에서 MRR 4.5%p의 성능 향상을 확인했다. 하지만, 아직 생성된 대화가 불안정한 경우가 존재하고 부자연스러운 대화가 만들어지는 경우가 존재한다. 따라서, 이러한 생성 과정에서 오류를 찾아내고 정제하거나 수정하는 방법에 대해 추가적으로 연구할 예정이다.

## Acknowledgement

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-02068, 인공지능 혁신 허브 연구 개발)

## 참고문헌

- [1] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Alpaca: A strong, replicable instruction-following model," *Stanford Center for Research on Foundation Models*. <https://crfm.stanford.edu/2023/03/13/alpaca.html>, Vol. 3, No. 6, p. 7, 2023.
- [2] Y. Anand, Z. Nussbaum, B. Duderstadt, B. Schmidt, and A. Mulyar, "Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo," *GitHub*, 2023.

- [3] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient finetuning of quantized llms,” *arXiv preprint arXiv:2305.14314*, 2023.
- [4] H. Ko, K. Yang, M. Ryu, T. Choi, S. Yang, Jiwung Hyun, and S. Park, “A technical report for polyglot-ko: Open-source large-scale korean language models,” 2023.
- [5] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [6] N. Houlsby, A. Giurghi, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-efficient transfer learning for nlp,” *International Conference on Machine Learning*, pp. 2790–2799, 2019.
- [7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.
- [8] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel, “Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning,” *Advances in Neural Information Processing Systems*, Vol. 35, pp. 1950–1965, 2022.
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, Vol. 30, 2017.
- [10] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, and S. Weinbach, “GPT-NeoX-20B: An open-source autoregressive language model,” *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 95–136, May 2022. [Online]. Available: <https://aclanthology.org/2022.bigscience-1.9>
- [11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in Neural Information Processing Systems*, Vol. 35, pp. 24 824–24 837, 2022.
- [12] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, “Dense passage retrieval for open-domain question answering,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.
- [13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

<p>▶ 대화 주제: 오아시스</p> <p>user: 오아시스라는 밴드 아세요? 유명한 밴드인가요?</p> <p>agent: 그럼요. 세계적으로 유명한 밴드잖아요.</p> <p>user: 그래요? 왜 그렇게 유명한 거예요?</p> <p>agent: 인기가 절정에 이를 때 내놓은 오아시스의 3집 《Be Here Now》(1997)가 영국 차트 1위, 미국 차트 2위에 올랐고, 발매 첫날에만 거의 50만장의 판매고를 기록하며 영국 역사상 가장 빨리 팔린 앨범이 되었거든요.</p> <p>user: 와, 정말 대단하네요. 근데 오아시스라는 이름은 누가 지은 거예요?</p> <p>agent: 리암 갤러거가 지었다고 알고 있어요. 인스파이럴 카펫츠의 투어 포스터에서 따왔다고 하더라고요.</p> <p>user: 그렇군요. 그럼 오아시스의 음악적 방향성은 뭐예요?</p> <p>agent: 단순함을 그들의 음악적 방향으로 잡았다고 알고 있어요. 본헤드는 바레코드만을, 킥시는 베이스의 루트음만을, 맥캐롤은 기본리듬만을 연주하도록 했어요. 앰프는 사운드가 찌그러질 때까지 켜두었대요.</p> <p>user: 아, 그런 깊은 뜻이 있었군요. 그럼 오아시스는 언제부터 활동을 시작한 거예요?</p> <p>agent: 오아시스는 1991년부터 활동을 시작했어요.</p> <p>user: 와, 정말 오래된 밴드네요. 그럼 오아시스는 언제 해체된 거예요?</p> <p>agent: 1992년에 해체되었어요.</p> <p>user: 그렇군요. 저는 오아시스가 아직도 활동하고 있는 줄 알았어요.</p> <p>agent: 하하, 그러셨군요. 오아시스가 해체된 이후에도 멤버들은 각자의 길을 걸었어요.</p> <p>user: 아, 그래요? 그럼 오아시스가 해체된 후에 발표한 곡 중에 영국 차트 10위권에 진입한 곡이 있나요?</p> <p>agent: 네, 있어요. 세 번째 싱글인 Live Forever가 처음으로 영국 차트 10위권에 진입했어요. user: 와, 정말요? 어떤 곡이길래 10위권에 진입했을까요?</p> <p>agent: Live Forever라는 곡이에요. 최고의 한 해였지만 실패없는 라이브 공연과 녹음 스케줄로 인해 밴드가 점점 지쳐갔고, 결국 1994년 9월 로스앤젤레스 공연에서 필로폰에 취한 리암이 미국 관중들에게 무례한 발언을 하고 멤버런으로 노엘을 때리는 등 공연이 난장판으로 바뀌는 사태가 벌어졌다는 내용이에요.</p> <p>user: 그렇군요. 오아시스가 해체되고 나서 블러와 라이벌 구도가 형성되었다고 하던데, 실제로 그랬나요?</p> <p>agent: 맞아요. 1995년에 블러와 오아시스가 같은 날 새 싱글을 발표하면서 브릿팝 전쟁을 촉발시켰어요.</p> <p>user: 그럼 오아시스와 블러가 라이벌 구도를 형성하기 전에는 누가 있었나요?</p> <p>agent: 비틀즈와 더 잼이 있었죠. 하루에 두 곡을 연주한 세 번째 밴드였어요.</p> <p>user: 아, 비틀즈와 더 잼이 있었군요. 그럼 오아시스는 언제 에비 로드 스튜디오에서 앨범을 녹음했나요?</p> <p>agent: 1996년 말부터 1997년 초까지요.</p> <p>user: 와, 비틀즈와 더 잼 이후로 영국에서 가장 빠른 판매고를 기록한 앨범이라니 대단하네요. 그 당시에 오아시스의 인기가 어느 정도였나요?</p> <p>agent: 발매 첫 날에만 423,000장이 팔렸어요. 영국 앨범 차트 2위에 올랐고요.</p> <p>user: 와, 비틀즈 이후로 영국에서 가장 빠른 판매고를 기록한 앨범이라니 정말 대단하네요. 오아시스의 다른 앨범들도 인기가 많았나요?</p> <p>agent: 네, Be Here Now라는 앨범도 인기가 많았어요.</p> <p>user: 그랬군요. 그럼 오아시스 멤버들 사이에 불화가 있었나요?</p> <p>agent: 네. 리엄이 제대로 노래할 수 있도록 녹음 기간에는 음주 및 마약 금지 명령을 내렸는데, 아더스가 이를 위반해서 노엘과의 사이에 갈등이 빚어졌어요.</p> <p>...</p>
---

표 3. 대화 생성 예제